A Secure and Trusted Fog Computing Approach based on Blockchain and Identity Federation for a Granular Access Control in IoT Environments

Samia EL HADDOUTI
ENSIAS
Mohamed V University in Rabat

Mohamed Dafir ECH-CHERIF EL KETTANI
ENSIAS
Mohamed V University in Rabat

Abstract—**Fog computing is a new computing paradigm that is an extension of the standard cloud computing model, which can be adopted as a cost effective strategy for managing connected objects , by enabling real-time computing and communication for analytical and decision making. Nonetheless, even though Fog-based Internet of Things networks optimize the standard architecture by moving computing, storage, communication, and control decision closer to the edge network, the technology becomes open to malicious attackers and remains many business risks that are not yet resolved. In fact, access control, privacy as well as trust risks present major challenges in Internet of Things environments based on Fog computing due to the large scale distributed nature of devices at the Fog layer. In addition, the traditional authentication methods are not adequate in Fog-based Internet of Things contexts since they consume significantly more computation power and incur high latency. To deal with these gaps, we present in this paper a secure and trusted Fog Computing approach based on Blockchain and Identity Federation technologies for a granular access control in IoT environments. The proposed scheme uses Smart Contract concept and Attribute-Based Access Control model to ensure the level of security and scalability required for data integrity without resorting to a central authority to make an access decision.**

Keywords*—*Access control; blockchain; fog computing; identity federation; IoT; smart contracts

I. Introduction

The Internet of Things (IoT) is fueling significant advances and smart services in various areas such as home automation, smart city, smart healthcare, intelligent transportation, etc; adding thereby value to businesses and increasing users convenience [1]. In fact, the rapid advance of communication and networking technologies, such as Bluetooth, WiFi, ZigBee, and GSM, enable connectivity among heterogeneous IoT subjects (e.g., smartphones, laptops, sensors, game consoles, etc.) to the Internet, which significantly accelerates data collection, aggregation and sharing in the IoT [2]. Yet, with expansion of IoT systems associated with big data from smart applications that require unlimited computing and storage resources, serious constraints with cloud-based solutions have been arisen due to real-time and reliable transport of enormous IoT traffic. Indeed, classical IoT infrastructures rely on centralized cloud computing paradigms to process and interpret large amounts of data sets, which include high latency and limited capacity with the increase of the latter. In addition, integrated Cloud Computing becomes a potential target for numerous security threats [3]. To address these technological gaps, Fog-based IoT network, which integrates network edge and cloud core, is recommended in recent years as a more effective solution to fulfill IoT requirements more positively [4], by extending the IoT network and expand its scope. The principle is making use of Fog Computing approach [5], [6] to offload network tasks (e.g., computing, storage, etc.), by moving computing and caching resources and analytical services closer to the edge network where data is generated [7]. Thus, data no longer needs to be sent in its entirety to data centers, which ultimately contributes to improving the quality of service. Fig. 1, illustrates the basic three-layer of a Fog-based IoT network.

While fog computing solves the aforementioned issues, new concerns arise in terms of security, privacy and trust that become more complex due to device heterogeneity, distributed management and mobility [8]. Moreover, to ensure mutual access control between Fog devices in such a distributed and unreliable environment, traditional authentication mechanisms, such as password-based authentication or certificate-based authentication methods, are no longer suitable. At the outset, several access control systems have been proposed but most of them are static for closed environment and they did not completely meet the dynamic Fog-based IoT requirements in term of scalability, data privacy and identity management. To deal with these downsides, the emerging Blockchain technology is seen as a new philosophy for building a truly decentralized, trust-less and secure access control structure for the Fog-based IoT networks [9]. From these perspectives, this work is introduces with the aim of overcoming the current Fog-based IoT limitations, by proposing a secure and trusted Fog Computing approach based on Blockchain and Identity Federation technologies for a granular access control in IoT environments. The given scheme aims to enhance typical Fog-based IoT networks as a result of a combination of strengths of two technologies: Identity Federation [10] to retrieve additional attributes from different Identity Providers ; within a trust circle; for granting access to an end user, and a consortium Blockchain to govern particular Smart Contracts to automate access control policies relied on Attribute-Based Access Control (ABAC) model [11], by providing far greater privacy and security and nullify the need for a third party. In particular, the proposed approach protects IoT devices by defining access control policies, which state the conditions of an end user's attributes to regulate the access to these IoT devices via a smart contract. The users' attributes are provided by Identity Providers members of an Identity Federation. In the aim to

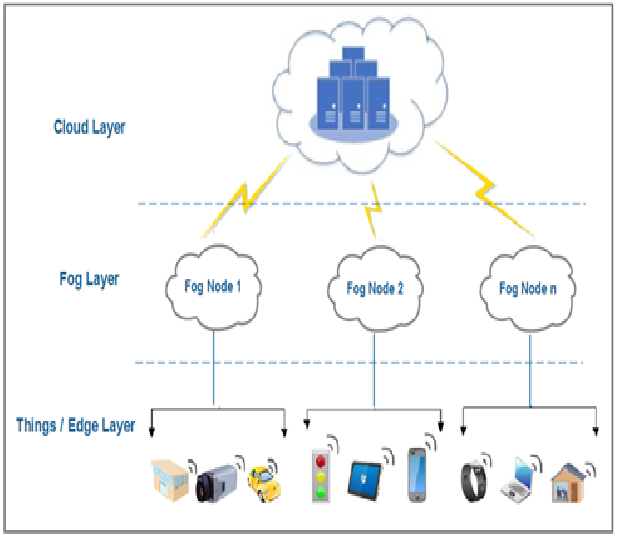


Fig. 1. Fog-based IoT Architecture.

elaborate a trusted environment, the nodes of the consortium Blockchain act as Service Providers within the given Identity Federation. The fog layer interacts with the given consortium Blockchain to check the relevant smart contract if a subscriber tries to apply the resources of a fog node.

The remainder of the paper is organized as follows: Section 2 reviews the existing approaches for the access control and authentication in Fog Computing environments. Section 3 presents the main security challenges in Fog-based IoT networks. Section 4 deals with properties of the Blockchain technology as a focal element for the proposed paradigm. Section 5 introduces the model by outlining its key features, followed by a detailed description of its components' architecture. Indeed, the section gives a system overview, architectural and main interactions among key components and actors. Section 6 is devoted to assessing the proposed approach. Finally, Section 7 is dedicated to conclusion and future work.

II. Related Works

Several works and initiatives have proposed various access control methods for Fog-based IoT networks.

In [12], Ibrahim et al. proposed a secure and mutual authentication model that allows any fog user to authenticate mutually with any fog server. This mutual authentication is under the registration authority (RA) at the cloud level. The main drawbacks of this approach, is the centralization of the RA that is considered as a Single Point of Failure. In fact, if the central RA is compromised, a negative impact would be felt in the whole system. In addition, the proposal did not consider the privacy aspect, by protecting the users' anonymity.

Similarly, Amor et al. [13] presented a solution of a mutual authentication between fog users and fog servers, by establishing a session key without disclosing user's real identity. The proposed scheme relies on various authentication methods and mechanisms such as bi-linear pairing, the elliptic curve discrete logarithm problem and pseudonym-based cryptography to enhance security aspects. However, the centralization feature of the given approach remains as a serious problem to deal with.

In [14], Imine et al. introduced an authentication method based on Blockchain technology and secret sharing technique to verify the authenticity of any fog node in the architecture, and to allow fog nodes to establish mutual authentication with each other. The major weakness of this method was its relationship with a centralized cloud. In fact, if the latter is compromised, there would be a negative impact on the whole system.

In recent years, there have several initiatives whose work was based on Blockchain and Identity Management for Fog-based IoT technology [15],[16],[17],[18]. Most of the features of these works are satisfied by our proposal. Furthermore, the latter is specially designed to added more features that enhance security and privacy aspects, by introducing Identity Federation technology as a crucial brick to ensure trusted interactions between different stakeholders to aggregate user's attributes from different Identity Providers for granting access to the requested IoT devices. In addition, the adoption of Blockchain technology, as a decentralized and distributed network, may be the most suitable environment for carrying out the authentication and authorization processes; through a smart contract; without resorting to a central authority and enhancing thereby the security and trustworthiness aspects. Thus, the proposed scheme will certainly pave the way to a wider adoption of the proposed paradigm by different industries.

III. Security Challenges in Fog-based IoT Networks

There have been immense efforts in recent years to cope with security issues in Fog-based IoT environments at different levels of gateways; though Identity Management, authentication and access control are the pillar security features that play a major role in establishing trust between Fog-based IoT components, by preventing malicious objects being easily connected to the network.

A. *Identity Management*

Generally speaking, Identity Management (IdM) aims to facilitate the management of identities in the digital world by decreasing extra administration costs. An integrated system that is in charge of ensuring IdM process is known as an Identity Management System (IdMS). It is generally made up of the following bricks [19]: *(i) Identity Provider (IdP)*, the structure that creates, manages, and maintains digital identities. Likewise, it generates assertions about identity attributes. *(ii) Service Provider (SP)*, it is also known as a Relying Party, corresponding with organizations that are providing resources and services to end users. *(iii) Control Party* refers typically to a regulatory body that uses identity information for investigations and access monitoring. The IdM models are mainly classified as conventional or isolated, centralized, federated and user centered [20],[21]. Nevertheless, Identity Federation model is more appropriate for ensuring a trust context between the different stakeholders; by optimizing the exchange of information related to user authentication on the basis of the establishment of agreements between IdPs and SPs [22].

There are currently several frameworks and standardization initiatives of IdMS in different phases of developments, and each of them has its own distinguishing features. In [23], the authors present a comparative analysis of the most popular IdMSs against a set of identity requirements. It is worth emphasizing the fact that even when there are many IdMS proposals in the literature; nevertheless; they do not meet the fog paradigm requirements. In this sense, establishing an IdM approach in the Fog-based IoT networks can be a challenging task to get a successful set up of an appropriate IdMS, which should take into account the highly dynamic network conditions expected in fog computing contexts and the large amount of computing resources required for a given operation.

B. Authentication

To access protected resources and services, entities need to be identified and authenticated as a part of information security. Authentication is the process by which a legitimate entity (e.g., a person, an organization or a device) proves a claim about holding specific identities. There are a variety of methodologies that can be used to authenticate entities. Existing authentication schemes are generally built on three main concepts that are based on the following [24]: *(i) Something an entity knows*; known as a knowledge-based authentication, this method involves the transmission of a secret, which is specific to an individual, by means of a password, code word, Personal Identification Number (PIN) and the like [25]. Despite its traditional and wide use, the level of risk protection afforded by these schemes is far from being adequate to ensure the required security level [26]. *(ii) Something an entity has*; is based on the possession of physical or digital private objects, referred to generally as a token, that the end user has. Examples of something an entity possesses include, among many others, digital certificates, smart cards, tokens and so on [27]. Although the security enhancement provided by this method, is useless in uncontrolled environments where a valid token may have been stolen. *(iii) Something an entity is*; referred to as a biometric-based authentication. This approach allows the authentication of entities based on either the human physiology or behavioural characteristics including fingerprint, iris, retinal, hand geometry, facial voice recognition, etc [28].

There are several approaches that have been started to implement authentication on IoT. However, traditional authentication schemes that exist in the web world will not be directly effective in fog computing due to the requirements of large computing power and real-time processing. Hence, new authentication techniques in fog computing have been proposed, each one with its strengths and weaknesses [29].

C. Access Control Models

After a successful authentication of an entity, an access control is required as the core of information security and shared data protection. It states policies and measures by which a Relaying Party determines whether an already authenticated entity has sufficient privileges to access the requested resource, and thus limits the actions that the legitimate entity can perform in such a way that only authorized access is possible. Access control models can be implemented in many places and at different levels [30].

1) Access Control Matrix: the Access Control Matrix (ACM) was the first theoretical access control model that defines access permissions between specific subjects and objects [31]. In this model, an access matrix, also known as a protection matrix, is designed with two-dimensional array, where the matrix rows are indexed by subjects, while matrix column are labelled by objects. This matrix acts as a lookup table for operating systems, where the context of each cell states the set of actions of a particular object that are allowed for a particular subject.

2) Role Based Access Control: David Ferraiolo and Rick Kuhn have elaborated on the RBAC model in 1992 [32], in which system permissions are assigned to users based on their roles so that security management costs are reduced. Indeed, the idea behind this model is that there will be fewer roles than users since users change frequently and roles do not. According to job functions, roles are created with privileges that are granted to users on the basis of their jobs or roles in the system. In other words, roles act as links between end users and resources. Despite its many benefits, the RBAC model could only be useful and suitable for organization whose trades and missions know little involvement.

3) Attribute Based Access Control: the Attribute Based Access Control (ABAC) model has been designed and developed ultimately to reduce the complexities of previous models within distributed and dynamic environments [33]. Under ABAC, the access to a protected resource is granted on the basis of the individual's attributes, of a resource, or of an environment. Access rules are created without the establishment of relationships between subjects and objects, which significantly increases the flexibility feature that is actually required in modern applications based on the emergence of the Service Oriented Architecture (SOA). For expressing access control policies, ABAC implementations are based on the eXtensible Access Control Markup Language (XACML) developed by the Advancing Open Standards for the Information Society (OASIS) [34]. Even though there are particular advantages of ABAC, a consensus definition of this approach is needed, and work remains to be done in assuring attribute accuracy and reliability.

IV. Blockchain Technology: Features and Working Principles

A. Overview

Blockchain [35] is a tamper resistant distributed database " Distributed Ledger" of recording transactions occurring within a network without the need for a central authority or third party (i.e., a bank, company, or government). A Blockchain contains a set of blocks, and every block contains a hash of the previous block, creating a chain of blocks from the *genesis* block to the current block as depicted in Fig. 2.

In 2008, Blockchain technology was combined with other computing concepts to create modern cryptocurrencies, and the first such Blockchain based cryptocurrency was Bitcoin [36]. The latter, along with certain cryptographic mechanisms, stores information representing electronic payments that are attached to digital addresses. Users use public and private keys to securely sign transactions within the system, allowing all participants to independently verify the validity of these

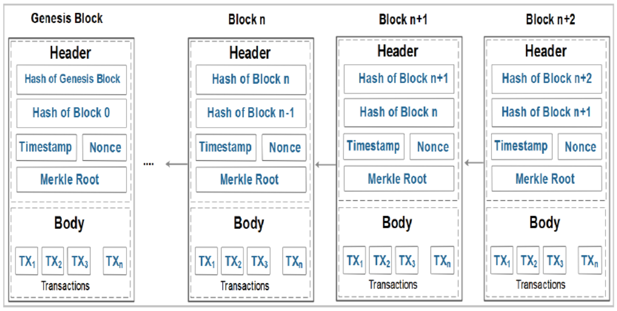


Fig. 2. High Level Structure of Blocks in a Blockchain System.

transactions through a consensus algorithm. After Bitcoin's success and growing visibility since its launch, Blockchain applications are gaining massive momentum in the last few years, and are wide used in fields of supply chain, financial, medical, IoT, and so on, where extensive research attention has been received [37].

B. Classification of Blockchain

Broadly, Blockchains are classified into three categories [38]:

1) Public Blockchain: is also termed *permissionless* Blockchain, and are perfectly represent decentralized systems, where everyone is allowed to participate in the network by reserving rights to publish blocks, access contents, maintain a copy of the distributed ledger, and participate in the validation of new blocks with the same authority like other participants. The conception of public Blockchain aims to host a large number of anonymous peers, which makes the tamper of its contents too costly, and thus the immutability and security of transactions are kept intact. However, in terms of infrastructure, public Blockchain require significant resources with more energy and power for their function and achieving validation consensus, which may also impact the speed of transactions.

2) Private Blockchain: unlike public Blockchain, private Blockchain are *permissioned*, by restricting members that can participate in the network. The access control is entrusted to one entity, and blocks are published by delegated peers within the network. The number of transactions per second is increased since the number of peers is less in a private Blockchain, which also speed up the performance of transactions. However, private Blockchain are not resistant enough to tampering, and are more prone to potential malicious behaviors.

3) Consortium Blockchain: Consortium Blockchain are also known as *Federated Blockchain*. They are hybrid of the previous two types, but they are closer to Private Blockchain since both of them are permissioned. The prime idea behind the adoption of a consortium Blockchain is to intensify the effect of cooperation in order to overcome the challenges of a particular industry. Indeed, by joining a consortium Blockchain, organizations will benefit from shared resources, decreased development time and cost, and increased consensus trust. This type of collaboration helps members of a consortium Blockchain to build business solutions with economies of scale.

C. Cryptographic Machanisms

Besides the hashing mechanism that is used to represent the current state of a Blockchain, by guarantying that no transactions in history can be tampered with, digital signatures [39] are another cryptographic concept that underpins the security of Blockchain technology. Indeed, in a Blockchain network, data transactions must be maintained by only approved parties. To that end, private keys are generated randomly and used in digital signatures required to spend transactions as proofs of ownership. Blockchain uses different types of cryptography including the Elliptic Curve Digital Signature Algorithm (ECDSA) [40] to authenticate transactions. However, it does not require digital certificates for its users to trust the integrity of the network because the Blockchain miners have already verified the transfer of digital values. Hence, they are not dependent on central authorities and servers as is the case with traditional Public Key Infrastructures (PKI) [41]. The whole process of a transaction signing in Blockchain is illustrated in Fig. 3.

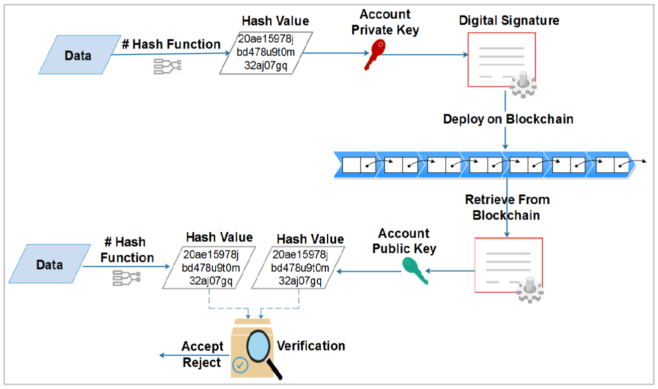


Fig. 3. Digital Signature Scheme on Blockchain.

D. Consensus Algorithms

Consensus algorithms are considered as the backbones and key elements in the working principles of Blockchains, by ensuring the network's security, integrity, and performance. Basically, consensus algorithms aim to reach a common assent and unanimity on the synchronized state of a distributed ledger among all participant peers, in order to find some measure of trust between unknown peers, achieving thus the integrity and reliability of information stored on Blockchain, while preventing tampering and the double spending problem in distributed environments. The most widely utilized consensus algorithms throughout public and private Blockchain infrastructures are:

1) Proof-of-Work (PoW) Algorithm: is the first consensus algorithm that was established with Blockchain. In PoW [42], peers compete against each other to be selected as a leader to add blocks to the chain, by performing computationally expensive amount of work to resolve a mathematical challenge in a predefined time (10 minutes for the Bitcoin Blockchain). Publishing new blocks is more widely known under the name of "mining".

2) Proof-of-Stake (PoS) Algorithm: following concerns due to the increase of energy consumption in Blockchains based on the PoW consensus, researchers have thought of alternatives for the said algorithm. Proof of Stake (PoS) [43] is one of the main candidates that have been proposed to solve the energy and resources expenditure problem created by the PoW. In fact, instead of the power of solving a computationally expensive puzzle, in the PoS participants have to proof the ownership of blocked money (i.e. stake) in their cryptocurrency wallets. The greater the stake, the more likely the peer is selected as a validator to generate the next block. Although the clear advantages of PoS over PoW to reach a consensus on a Blockchain network, PoS possesses several potential security issues. These include a lack of initial coin distribution, a threat to decentralization, and an increased chance of double-spending when forks occur due to identical node verification [44].

3) Delegated Proof-of-Stake (DPoS) Algorithm: in this algorithm [45], a group of nodes are elected by stakeholders to produce and publish blocks. These nodes are called producers or witnesses. The number of elected witnesses is defined in such a way that at least 50% of nodes trust there is enough decentralization. Typically, witnesses take turns generating a block within a fixed time interval. For each block generation, the corresponding witness is rewarded. However, in the case where a witness has not generated any block within the fixed schedule, it is removed from the elected group until its notification of the intention to start generating blocks again.

4) Practical Byzantine Fault Tolerance (PBFT) Algorithm: is specifically intended for permissioned Blockchains where the number of participants is usually lower than public Blockchains. By this virtue, reaching a consensus therefore does not require costly proofs. The PBFT is based on state machine replication approach [46]. It aims to reduce transmission errors, while introducing considerable optimizations that improve the response time of previous algorithms.

E. Smart Contract Concept

The concept of Smart Contracts was introduced by Szabo in 1997 [47], where he defined the Smart Contract as a computer code including terms and clauses of a traditional contract that is executing automatically. With the emergence of Blockchain technology, this approach has become feasible and viable. Indeed, the combination of Smart Contracts with Blockchain technology has changed the way businesses are currently done since “contracts” can be utilized and executed easily and quickly. As a matter of course, this innovative approach might replace traditional legal and economic contracts that are enforced by centralized entities such as lawyers, insurance agencies, and banks. The execution of a Smart Contract within a Blockchain network does not require intermediaries to verify and validate its terms and clauses. From a technical point of view, a Smart Contract performs the function of carrying out transactions via the execution of a related code, with predefined rules, that is stored on a distributed ledger and is identified by a unique address. Deploying Smart Contracts has undoubtedly brought considerable benefits to business and customers as is already mentioned. Nevertheless, advantages of the adoption of Smart Contracts could not come up without challenges. In fact, security vulnerabilities can occur to make a series of attacks against any network possible in the case of existing bugs or loopholes in deployed codes, which become more complex to manage with the immutable feature of the Blockchain system [48].

V. Proposed Fog-based IoT Scheme

In this section, we present the detailed methodology of our proposed access control model based on Blockchain and Identity Federation technologies to enhance security, privacy and trust aspects within Fog-based IoT networks.

A. System Architecture

The proposed architecture of the system is depicted in Fig. 4. It consists of five main components and these are as follows:

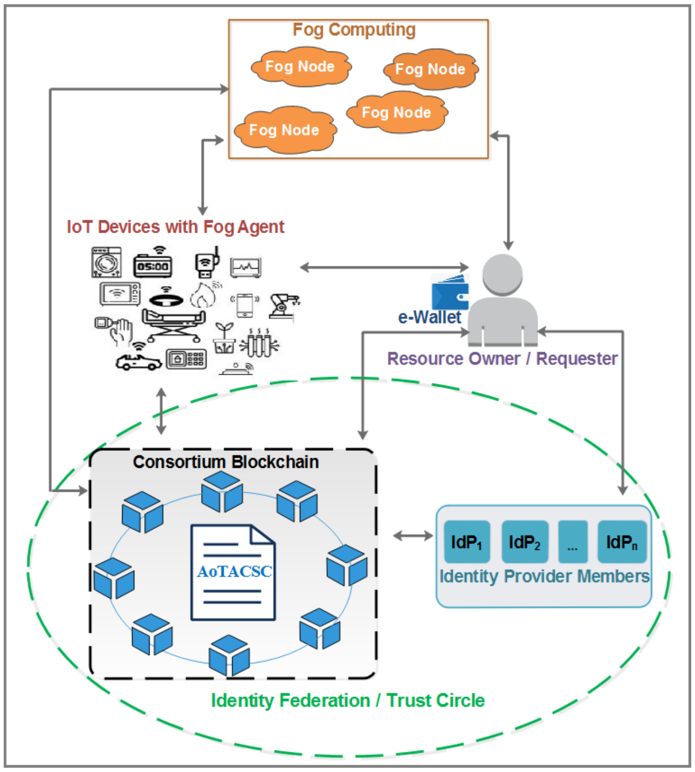

Fog Computing
Fog Node
Fog Node
Fog Node
Fog Node
IoT Devices with Fog Agent
05:00
e-Wallet
Resource Owner / Requester
Consortium Blockchain
AoTACSC
IdP1
IdP2
...
IdPn
Identity Provider Members
Identity Federation / Trust Circle

Fig. 4. Proposed System Architecture.

1) End Users: an end user can play the role of a resource owner who protects his resources by defining access over it, or a resource requester who aims to access protected resources. It is worth emphasizing that every end user has at least one electronic wallet that includes his credentials, addresses and all the keys needed to sign and validate transactions, and ask for resources access. In the proposed paradigm, we consider a wallet as a Distributed Application through which an end user could interact with the Blockchain platform to register his resources that need to be protected and to define his access control policies.

2) Fog Nodes: these nodes are hosting fog services, by providing a real-time execution of services at the edge of the network and performing effectively and efficiently computation and communication. However, unlike the previous proposed approaches of access control in Fog Computing in which access policies is usually conferred to fog nodes, thereby malicious intruders can gain the possibility of compromising these policies; the given proposed solution delegates the access control process to a consortium Blockchain. The motivation behind this approach is essentially to establish efficient access control policies by ensuring confidentiality, accountability and integrity. In this way, if an end user tries to apply the resources of a fog node and in order to make data driven decisions, the fog node retrieves securely information from the consortium Blockchain by triggering the execution of a Smart contract, which will generates an output with information including the authentication assertion in addition to the access profile that indicates the number of leased resources, their duration, and their scale authorized for the given end user.

3) Consortium Blockchain: is the focal point of the proposed architecture to ensure the access control for IoT devices. Indeed, this consortium Blockchain is made up of nodes able to successfully process authentication and authorization functions via a smart contract called *IoT Access Control Smart Contract (IoTACSC)*, which is a representation of the access control policies for each pair (resource, end user) on the basis of the ABAC model. Nodes of the Consortium Blockchain act as Service Providers within an Identity Federation and interact with end users, Identity Providers and fog computing nodes. The execution outcome of the IoTACSC is validated by all Blockchain nodes before being recorded in a distributed ledger that is obviously only shared, replicated and synchronized among the nodes of the Consortium Blockchain to increase consensus trust with economies of scale, thereby providing data security and network privacy in the Fog Computing environments. It is noteworthy that every fog node submits its IoTACSC onto the consortium Blockchain, and if an end user tries to request a resource of a fog node, the latter checks the relevant contract from the Consortium Blockchain.

4) Identity Providers: to enhance the privacy aspect and come into line with the typical approach of Identity Federation, digital identities of end users still remains the involvement of their home organizations via trusted IdPs. The latter provide identity attributes of end users to get fog resources access according to access control policies described in IoTACSC. As external entities, IdPs interacts with the IoTACSC through *Application Program Interfaces (API)*.

5) IoT Devices: these devices are in the form of wireless sensors constrained in their computational power and energy availability. Those limitations restrict them to be part of the consortium Blockchain, since being part of the Blockchain network implies keeping a copy of the Blockchain locally and a track of the network transactions. Nevertheless, all the devices are uniquely identified in the consortium Blockchain by creating automatically a public key for every device. Thus, each IoT device will have a unique identifier illustrated on the Access Control Policy. To interact with the consortium Blockchain and fog computing nodes, a fog agent may be deployed on access devices to help end users to request fog resources.

B. System Interactions

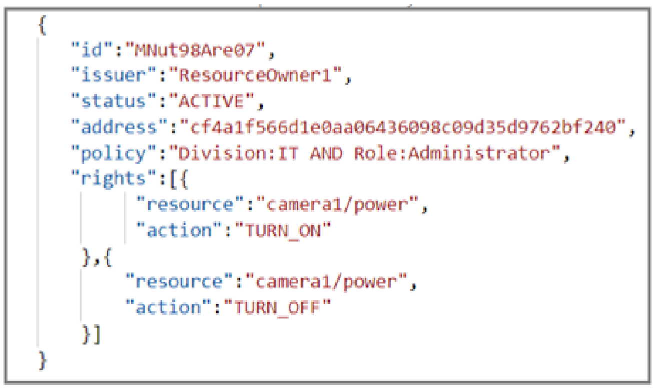
this section explains the different interactions between the different components of the proposed architecture. These interactions can be divided into two different phases:

1) Registration Phase: this phase relates to the establishment of access control policies regulating the access to IoT devices. Before putting forward the registration process, it should be noted that resource owners are the only entities with the ability to interact with the Consortium Blockchain in order to define new policies to access the relevant IoT device. Thus, a resource owner is always asked for authentication prior being able to interact with the nodes of permissioned Blockchain. The authentication protocol is composed of two processes, including registration and login. In essence, the identity of a resource owner within the consortium Blockchain is built up by binding his *Wallet's Public Key (WPubKey)* with a unique *user identifier (UsrID)*, and then uploading the said identity on the Consortium Blockchain in form of an identity transaction. The components of the latter are shown in Table I. The workflow of the registration phase is as follow: after a successful authentication, the resource owner is invited to register the resource under his control. After the registration of the IoT device, the resource owner has to outline how access is authorized to the resource device by defining an access policy with corresponding access rights to specify which group of resource requesters can perform what actions to the given IoT device. For that end, the given access policies have to state the conditions of attributes that need to be satisfied to grant the access resource. The predefined access policy is transtated to an AoTACSC, which is then broadcasted to all the nodes of the consortium Blockchain. These nodes reach agreement about the received smart contract and the validated data is added to the ledger of the Consortium. This registration flow is clearly illustrated through the sequence diagram in Fig. 6.

TABLE I. Components of an Identity Transaction

Information	Description
TransactionType	Identity
User	Resource Owner
UserID	Alice001
WPubKey	fcf4a1f566d1e0aa06436098c09d35d9762bf240
UserName	Alice001
Password	@lice001!!
Timestamp	The time the transaction occurs, i.e., 177131cee76
Signature	0xc3373d3bd1d4edc089001fd330920c303e 95c51b131c22bc91b2f9f9f56e0de9

2) Resource Access Phase: The resource requester attempts to access a protected resource managed by a fog node within fog computing. In this stage, we assume that the requester is already aware of the access control policy regulating the access to the protected device. Referring to Fig. 7, a detailed view illustrating the resource access phase is provided. The resource requester sends a resource request to the fog agent on IoT device. The latter search the relevant IoTACSC address on consortium Blockchain, then communicate the found address to the requester. The resource requester submits his access request through his wallet to the consortium Blockchain. The relevant transaction is broadcasted to all nodes that evaluate the transaction, by executing the IoTACSC already deployed by the owner of the requested resource. For that end, the resource requested is asked to select trusted IdPs that are members of the Identity Federation system, and are managing the attributes required by IoTACSC. The end user is then redirected to the chosen IdP for authentication. In the case of a successful authentication, Blockchain nodes send an attribute query combined with an authentication assertion to the given IdP, which prepares an attribute assertion and return the result to the consortium Blockchain. At this step, the end user may be asked to select another IdP, and the previous process is repeated till nodes get the set of attributes defined in IoTACSC. Afterword, based on a set of attribute assertions, the execution of IoTACSC makes an authorization decision about the end user's request. In fact, if it was successfully executed, the IoTACSC generates a secret access key and assigns an access token, which indicates the access rights for the resource requester. This access token is broadcasted to every node in the consortium Blockchain. Nodes reach agreement about the received access token, which is then recorded into the consortium Blockchain. An access token contains a unique identifier (ID), the address of the resource requester, the policy that must be satisfied, a list of access rights and the current status. An example of an authorisation token is illustrated in Fig. 5. The resource requester uses the authorization token to access the targeted resource. When a Fog node is receiving requests with an access token, it will check and verify its validity, by referring to the consortium Blockchain. If this access token was delivered by the IoTACSC corresponding to the IoT device, it allows access else it denies.



```
{
    "id":"MNut98Are07",
    "issuer":"ResourceOwner1",
    "status":"ACTIVE",
    "address":"cf4a1f566d1e0aa06436098c09d35d9762bf240",
    "policy":"Division:IT AND Role:Administrator",
    "rights":[{
            "resource":"camera1/power",
            "action":"TURN_ON"
     },{
           "resource":"camera1/power",
           "action":"TURN_OFF"
     }]
}
```

Fig. 5. Example of an Authorisation Token.

C. Design of IoT Access Control Smart Contract (IoTACSC)

As mentioned above, to realize automation, efficiency and credibility of transactions corresponding to access control process, the proposed approach consists of a smart contract (IoTACSC) which implements predefined access control policies; based on ABAC model; to control the access requests from subjects, by expressing conditions over a set of attributes paired to the latter. More broadly, IoTACSCs of the proposed system allow object owners to conduct a registration process by implementing access control policies. We designed the registration of IoT devices corresponding to access control policies as in Algorithm 1, which receives the identifier of IoT device ($Id_{\text{IoT\_Device}}$) as input, and returns an address of this IoT device ($Address_{\text{IoT\_Device}}$) and an address of the IoTACSC ($Address_{\text{IoTACSC}}$).

On the other hand, IoTACSCs endorse the permission decision process, by determining whether an end user is allowed to perform the access operation on an IoT device according to the exclusive access control policy deployed by the resource owner. We designed the permission decision policy as in Algorithm 2, which receives the identifier of resource requester (EU_Id), the address of the requested IoT device ($Address_{\text{IoT\_Device}}$) and the address of the IoTACSC ($Address_{\text{IoTACSC}}$) as input, and returns an access token ($Access\_TKN$), then the judgment result ("Grant" or 'Deny").

Algorithm 1: Registering a New IoT Device.

```
/* This algorithm translates an
   access control policy in form of
   IoTACSC and records the latter on
   the consortium Blockchain          */
Input : $Id_{\text{IoT\_Device}}$: is the identifier of a target
        device. $AC\_Policy_{\text{IoT\_Device}}$: is an Access
        Policy related to a target device.
Output: $Address_{\text{IoT\_Device}}$, $Address_{\text{IoTACSC}}$
Auth_Control$_{\text{BC}}$: Authentication Control at the
  Consortium Blockchain level
/* A boolean function that checks if
   a given Device Owner is well
   authenticated at the Consortium
   Blockchain                         */
DO_BCCheckAuthentication : REQ × 
  Auth_Control$_{\text{BC}}$= {true, false}
H: is a hash function.
/* A boolean function that makes a
   decision on whether an end user
   may access a device resource in a
   particular environment             */
$AC\_Policy\_Rules(eu, d_{\text{r}}, e) \leftarrow$
  $f(ATT(EU), ATT(D_{\text{r}}), ATT(E))$
if $DO\_BCCheckAuthentication = true$ then
    /* Generate an address of the IoT
       device                         */
    $H(IoT\_Device_{\text{PubKey}}) \leftarrow Address_{\text{IoT\_Device}}$
        /* Create an IoTACSC and
           generate its address       */
    $AC\_Policy(eu, d_{\text{r}}, e) \leftarrow$
      $ATT(EU) \wedge ATT(D_{\text{r}}) \wedge ATT(E)$
      $Address_{\text{IoTACSC}} \leftarrow$
      $add.IoTACSC(AC\_Policy(eu, d_{\text{r}}, e))$
else
    return a rejection notification
end
return $Address_{IoT\_Device}$, $Address_{IoTACSC}$
```

VI. Analysis and Evaluation

A. Potential Advantages

It is clear that the proposed approach can actually added significant and considerable values for Fog-besed IoT environments, by providing an access control layer, while promoting the decentralization aspect and preserving security and privacy requirements. Indeed, the main offered advantages of the proposed paradigm are highlighted as follows: noitemsep

- Access control processes are managed through IoTACSC, which conduct faster transactions at lower

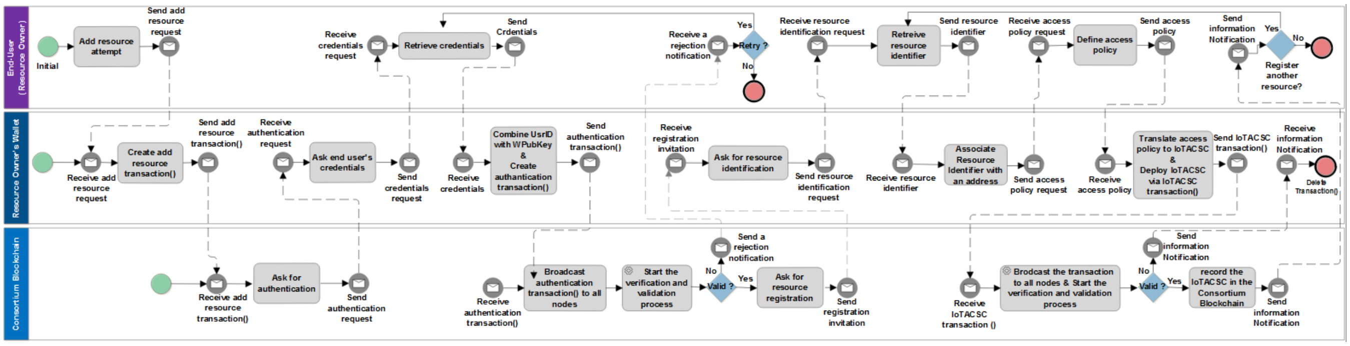

End-User (Resource Owner)
Initial
Add resource attempt
Send add resource request
Receive credentials request
Retrieve credentials
Send Crdentials
Receive a rejection notification
Retry ?
Yes
No
Receive resource identification request
Retreive resource identifier
Send resource identifier
Receive access policy request
Define access policy
Send access policy
Send information Notification
Yes
No
Register another resource?
Resource Owner's Wallet
Receive add resource request
Create add resource transaction()
Send add resource transaction()
Receive authentication request
Ask end user's credentials
Send credentials request
Receive credentials
Combine UsrID with WPubKey & Create authantication transaction()
Send authentication transaction()
Receive registration invitation
Ask for resource identification
Send resource identification request
Receive resource identifier
Associate Resource Identifier with an address
Send access policy request
Receive access policy
Translate access policy to IoTACSC & Deploy IoTACSC via IoTACSC transaction()
Send IoTACSC transaction()
Receive information Notification
Delete Transaction()
Consortium Blockchain
Receive add resource transaction()
Ask for authentication
Send authentication request
Receive authentication transaction()
Broadcast authentication transaction() to all nodes
Start the verification and validation process
Valid ?
No
Yes
Send a rejection notification
Ask for resource registration
Send registration invitation
Receive IoTACSC transaction ()
Brodcast the transaction to all nodes & Start the verification and validation process
Valid ?
No
Yes
Send information Notification
record the IoTACSC in the Consortium Blockchain
Send information Notification

Fig. 6. Registration Workflow.

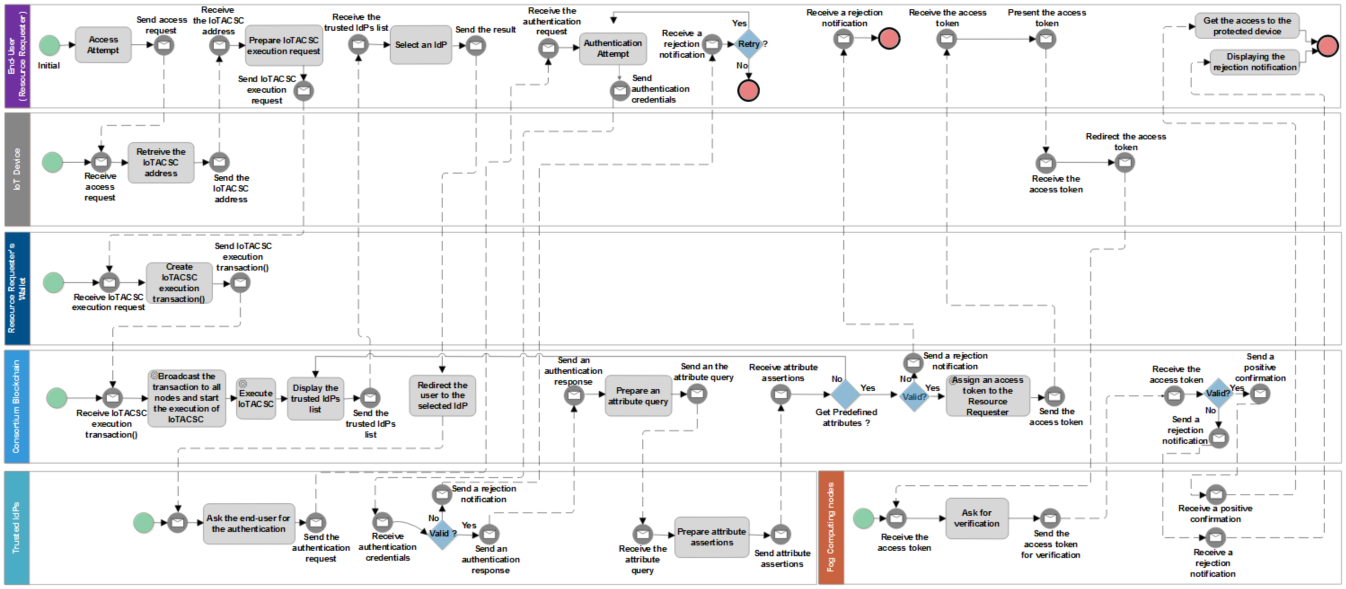

End-User (Resource Requester)
Initial
Access Attempt
Send access request
Receive the IoTACSC address
Prepare IoTACSC execution request
Send IoTACSC execution request
Receive the trusted IdPs list
Select an IdP
Send the result
Receive the authentication request
Authentication Attempt
Send authentication credentials
Receive a rejection notification
Retry ?
Yes
No
Receive a rejection notification
Receive the access token
Present the access token
Get the access to the protected device
Displaying the rejection notification
IoT Device
Receive access request
Retreive the IoTACSC address
Send the IoTACSC address
Receive the access token
Redirect the access token
Resource Requester's Wallet
Receive IoTACSC execution request
Create IoTACSC execution transaction()
Send IoTACSC execution transaction()
Consortium Blockchain
Receive IoTACSC execution transaction()
Broadcast the transaction to all nodes and start the execution of IoTACSC
Execute IoTACSC
Display the trusted IdPs list
Send the trusted IdPs list
Redirect the user to the selected IdP
Send an authentication response
Prepare an attribute query
Send an the attribute query
Receive attribute assertions
Get Predefined attributes ?
No
Yes
Valid?
No
Yes
Send a rejection notification
Assign an access token to the Resource Requester
Send the access token
Receive the access token
Valid?
Yes
No
Send a positive confirmation
Send a rejection notification
Trusted IdPs
Ask the end-user for the authentication
Send the authentication request
Receive authentication credentials
Valid ?
No
Yes
Send a rejection notification
Send an authentication response
Receive the attribute query
Prepare attribute assertions
Send attribute assertions
Fog Computing nodes
Receive the access token
Ask for verification
Send the access token for verification
Receive a positive confirmation
Receive a rejection notification

Fig. 7. Resource Access Workflow.

costs, improve its secrecy and achieve greater control over exchanged data.

- By combining the features of the Identity Federation and federated Blockchain, it is possible to take advantage of the benefits of both technologies regarding the trust management.
- The adoption of the consortium Blockchain instead of a public Blockchain speeds up transaction handling, by limiting the number of participant nodes needed to validate transactions during the permission decision process, without leading to the risk of centralization;
- The proposed approach preserves and promotes the privacy requirement while linking identities across different trust domains. In fact, identities are always maintained by related home IdPs. Only the corresponding attributeIDs are stored on the Consortium Blockchain and their values are known to home IdPs managing identity attributes.
- Since the majority of fog nodes will not be able to store Blockchain information due to their constrained nature, our architecture does not include fog nodes in the Blockchain. Consequently, the given may be easily and widely adopted by reaching a wider audience.

B. Open Issues

Although several advantages of the proposed scheme led us to positive statements, there are some open problems which are interesting to investigate further. In fact, it is wise to be aware of the following limitations: noitemsep

- The proposed model presents latency issues due to the time taken to mining transactions on Consortium Blockchain.
- As every node needs to process and verify transactions and maintain the updated copy of the distributed ledger, the inter-node latency increases. Thus, the system needs to improve the default consensus algorithm, while maintaining security and avoiding double spending issues;
- It is harder for the nodes to maintain the full copy of the ledger with the increase of data. Thus the nodes should be equipped with powerful hardware.

Algorithm 2: Resource Access Process

```
Input  : $EU\_Id$: is the identifier of an Resource
         Requester. $Address_{\text{Resource}}$: is the address of
         the target device resource. $Address_{\text{IoTACSC}}$:
         is the address of IoTACSC.
Output: $Access\_TKN$: is an access token.
         ACD: an access control decision regarding a
protected device resource {Grant, Deny}

$P_{\text{D}}$:is a set of permissions associated to atarget device
  D.
$P_{\text{D}}$ExpectedAttributes:$P_{\text{D}} \rightarrow$ {Att}, a function
  returning the set of attributes related to the $P_{\text{D}}$.
$IDF_{\text{F}}$: is a set of IdPs members of an Identity
  Federation F.
$AuthAssertion(UId)_{IdP_{\text{i}}}$: is an authentication
  assertion of an end user (UId) at a home $IdP_{\text{i}}$.
/* A set of home IdPs supposed to
   provide attributes required by $P_{\text{D}}$
   */
ExpectedIdPs: {$P_{\text{D}}$ExpectedAttributes}$\rightarrow$ {IdPs}
   /* The set of aggregated
   attributes from different IdPs   */
aggregatedAttributesList $\leftarrow$ {}
foreach $IdP_i \in ExpectedIdPs$ do
    if $AuthAssertionA(UId)_{IdP_i}$ = $OK$ then
        /* Get attribute values
           corresponding to $attributeIDs$
           for $UId$ according to
           $P_{\text{D}}ExpectedAttributes$          */
        foreach
         $attributeID \in \{P_D ExpectedAttributes\}$ do
            $aggregatedAttributesList.append$
            $(IdP_{\text{i}}.getAttributeValue$
             $(AuthAssertion(UId)_{\text{IdP}_{\text{i}}}, attributeID))$
        end
        return $ExpectedAttributes$
    end
end
/* Generate a token $ACC\_TKN$       */
if $aggregatedAttributesList.hasReqAttributeID()$
  then
    $(result$,ACC_TKN) $\leftarrow$ $returnResult()$
else
    return $a\ rejection\ notification$
end
/* Access Decision                   */
if $IoTACSC_D.CheckIoTACSC_D(ACC\_TKN) =$
  $Allow$ then
    ACD $\leftarrow$ ”Grant”
else
    ACD $\leftarrow$ ”Deny”
end
return $ACD$
```

VII. Conclusion and Future Work

Fog-based IoT networks have been designed as a solution to efficiently manage the resource continuum from the edge up to the cloud, by providing enormous opportunities and also bringing remarkable challenges. Access control considered one of the main challenges introduced by IoT devices and fog nodes that are not able to protect themselves due to their limited processing and storage capabilities. Several models and approaches have been proposed with a lack of scalability and vulnerabilities to cyberattacks.

In this paper, we propose a new scheme that combines the ABAC model with Blockchain and Identity Federation technologies. The proposed approach can solve the introduced issues in the open Fog-based IoT environments. Indeed, by adopting the Smart Contract principle, our proposed model provides secure, dynamic, reliable and scalable access control and authentication policies. The operating principle relies on the conception of an IoT Access Control Smart Contract deployed on a consortium Blockchain, where an object owner defines an access control policy managing the exploitation of the given object, by referring to a set of attribute aggregated from different Identity Providers belonging to an Identity Federation, enhancing thereby the trust management.

At this step, we have designed the approach. In our future work, we will implement the latter to demonstrate the feasibility of using Blockchain technology to manage access control process for IoT devices within Fog Computing environments. Moreover, we intent to design a lightweight consensus algorithm, which is expected to be more effective and well suited for the philosophy of the conceptual model, reducing the computational power and latency.

References

[1] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, “Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges,” *IEEE wireless communications*, vol. 24, no. 3, pp. 10–16, 2017.

[2] L. Da Xu, W. He, and S. Li, “Internet of things in industries: A survey,” *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[3] M. Kazim and S. Y. Zhu, “A survey on top security threats in cloud computing,” *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 3, 2015.

[4] H. R. Abdulqadir, S. R. Zeebaree, H. M. Shukur, M. M. Sadeeq, B. W. Salim, A. A. Salih, and S. F. Kak, “A study of moving from cloud computing to fog computing,” *Qubahan Academic Journal*, vol. 1, no. 2, pp. 60–70, 2021.

[5] Y. PAN and G. LUO, “Cloud computing, fog computing, and dew computing,” *ZTE COMMUNICATIONS*, vol. 15, no. 4, pp. 1–2, 2017.

[6] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, “Fog computing: Focusing on mobile users at the edge,” *arXiv preprint arXiv:1502.01815*, 2015.

[7] J. He, J. Wei, K. Chen, Z. Tang, Y. Zhou, and Y. Zhang, “Multitier fog computing with large-scale iot data analytics for smart cities,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 677–686, 2017.

[8] P. Zhang, M. Zhou, and G. Fortino, “Security and trust issues in fog computing: A survey,” *Future Generation Computer Systems*, vol. 88, pp. 16–27, 2018.

[9] A. Baouya, S. Chehida, S. Bensalem, and M. Bozga, “Fog computing and blockchain for massive iot deployment,” in *2020 9th Mediterranean Conference on Embedded Computing (MECO)*. IEEE, 2020, pp. 1–4.

[10] D. Rountree, *Federated identity primer*. Newnes, 2012.

[11] C. Hu *et al.*, “Nist special publication 800-162. guide to attribute based access control (abac) definition and considerations. national institute standards and technology,” 2016.

[12] M. H. Ibrahim, “Octopus: An edge-fog mutual authentication scheme.” *Int. J. Netw. Secur.*, vol. 18, no. 6, pp. 1089–1101, 2016.

[13] A. B. Amor, M. Abid, and A. Meddeb, "A privacy-preserving authentication scheme in an edge-fog environment," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2017, pp. 1225–1231.

[14] Y. Imine, D. E. Kouicem, A. Bouabdallah, and L. Ahmed, "Masfog: an efficient mutual authentication scheme for fog computing architecture," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2018, pp. 608–613.

[15] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, "Fairaccess: a new blockchain-based access control framework for the internet of things," *Security and communication networks*, vol. 9, no. 18, pp. 5943–5964, 2016.

[16] G. Liu, J. Wu, and T. Wang, "Blockchain-enabled fog resource access and granting," *Intelligent and Converged Networks*, vol. 2, no. 2, pp. 108–114, 2021.

[17] M. J. Baucas, P. Spachos, and K. N. Plataniotis, "Public key reinforced blockchain platform for fog-iot network system administration," *IEEE Internet of Things Journal*, 2021.

[18] Y. Zhang, R. Nakanishi, M. Sasabe, and S. Kasahara, "Combining iota and attribute-based encryption for access control in the internet of things," *Sensors*, vol. 21, no. 15, p. 5053, 2021.

[19] E. Bertino and K. Takahashi, *Identity management: Concepts, technologies, and systems*. Artech House, 2010.

[20] A. Jøsang and S. Pope, "User centric identity management," in *AusCERT Asia Pacific information technology security conference*, vol. 77. Citeseer, 2005.

[21] A. Bhargav-Spantzel, J. Camenisch, T. Gross, and D. Sommer, "User centricity: a taxonomy and open issues," *Journal of Computer Security*, vol. 15, no. 5, pp. 493–527, 2007.

[22] M. Gaedke, J. Meinecke, and M. Nussbaumer, "A modeling approach to federated identity and access management," in *Special interest tracks and posters of the 14th international conference on World Wide Web*, 2005, pp. 1156–1157.

[23] S. E. Haddouti and M. D. E.-C. E. Kettani, "Towards an interoperable identity management framework: a comparative study," *arXiv preprint arXiv:1902.11184*, 2019.

[24] L. Müller, "Authentication and transaction security in e-business," in *IFIP International Summer School on the Future of Identity in the Information Society*. Springer, 2007, pp. 175–197.

[25] A. Conklin, G. Dietrich, and D. Walz, "Password-based authentication: a system perspective," in *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*. IEEE, 2004, pp. 10–pp.

[26] Z. Zhao, Z. Dong, and Y. Wang, "Security analysis of a password-based authentication protocol proposed to ieee 1363," *Theoretical Computer Science*, vol. 352, no. 1-3, pp. 280–287, 2006.

[27] S. Zhao and W. Hu, "Improvement on otp authentication and a possession-based authentication framework," *International Journal of Multimedia Intelligence and Security*, vol. 3, no. 2, pp. 187–203, 2018.

[28] V. Matyas and Z. Riha, "Toward reliable user authentication through biometrics," *IEEE Security & Privacy*, vol. 1, no. 3, pp. 45–49, 2003.

[29] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for internet of things applications: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2017.

[30] E. Bertino, S. Das, K. Kant, and N. Zhang, "Policies, access control, and formal methods," 2012.

[31] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman, "Protection in operating systems," *Communications of the ACM*, vol. 19, no. 8, pp. 461–471, 1976.

[32] R. Sandhu, D. Ferraiolo, and R. Kuhn, "American national standard for information technology–role based access control," *ANSI INCITS*, vol. 359, p. 1, 2004.

[33] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone *et al.*, "Guide to attribute based access control (abac) definition and considerations (draft)," *NIST special publication*, vol. 800, no. 162, pp. 1–54, 2013.

[34] T. Moses, "Extensible access control markup language (xacml) version 2.0," *OASIS standard*, 2005.

[35] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," *arXiv preprint arXiv:1906.11078*, 2019.

[36] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[37] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE transactions on knowledge and data engineering*, vol. 30, no. 7, pp. 1366–1385, 2018.

[38] G. Hileman and M. Rauchs, "2017 global blockchain benchmarking study," *Available at SSRN 3040224*, 2017.

[39] A. Shamir, "New directions in croptography," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2001, pp. 159–159.

[40] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," *International journal of information security*, vol. 1, no. 1, pp. 36–63, 2001.

[41] J. Buchmann, E. Karatsiolis, A. Wiesmaier, and E. Karatsiolis, *Introduction to public key infrastructures*. Springer, 2013, vol. 36.

[42] B. Sriman, S. Ganesh Kumar, and P. Shamili, "Blockchain technology: Consensus protocol proof of work and proof of stake," in *Intelligent Computing and Applications*. Springer, 2021, pp. 395–406.

[43] F. Saleh, "Blockchain without waste: Proof-of-stake," *The Review of financial studies*, vol. 34, no. 3, pp. 1156–1190, 2021.

[44] P. Vasin, "Blackcoin's proof-of-stake protocol v2," *URL: https://blackcoin. co/blackcoin-pos-protocol-v2-whitepaper. pdf*, vol. 71, 2014.

[45] D. Larimer, "Dpos consensus algorithm-the missing white paper," *Bitshare whitepaper*, 2017.

[46] F. B. Schneider, "The state machine approach: A tutorial," *Fault-tolerant distributed computing*, pp. 18–41, 1990.

[47] N. Szabo, "Formalizing and securing relationships on public networks," *First monday*, 1997.

[48] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (sok)," in *International conference on principles of security and trust*. Springer, 2017, pp. 164–186.